REMARKS

Applicant has carefully considered the Office Action dated November 5, 2002 and the references cited therein. Applicant respectfully requests reexamination and reconsideration of the application.

Claim 27 has been amended to correct the spelling of a term therein. This amendment has not been made to distinguish over any reference of record and no narrowing of any corresponding equivalents to which the amended limitation(s) or claim(s) is/are entitled is intended by these amendments.

Claims 1-30 stand rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent No. 6,263,379, Atkinson et al., hereafter Atkinson, in view of U.S. Patent No. 6,363,433, Nakajima. In setting for the rejection of claim 1, the Examiner has alleged that Atkinson discloses the claimed subject matter, however, the Examiner has admitted that Atkinson does not teach substituting a moniker object for the distributed object. Instead, the Examiner is relying on Nakajima to supply such teaching, alleging the Nakajima discloses a first stream object (MKParseDisplayName, line 46 column 5), which automatically substitutes the moniker object (76, Fig. 3) for the distributed object (72, Fig. 3) when the distributed object is streamed out (arrows going from 38 to 82, Fig. 3) from the memory (38, Fig. 3) to the local storage (82, Fig. 3). The Examiner further asserts that it would have been obvious to apply the teachings of Nakajima to the system of Atkinson because the moniker object can be used in interfacing between the browser and the extension.

In response to the arguments set forth by the Applicant in the previous response, the Examiner has stated:

> Applicant argued that MKParseDisplayName is not a stream object (remarks, lines 10-11 first paragraph page 5). In response, in object-oriented programming, an object is a self-contained entity that consists of both data and procedures to manipulate the data. As stated in the rejection of claim 1, MKParseDisplayName substitutes the moniker object for the distributed object when the distributed object is streamed out from

2

the memory to the local storage. Therefore, the reference meet the limitation as broadly claimed.

Applicant respectfully disagrees with the Examiner's analogy. The fact that the MKParseDisplayName function is allegedly an object that consists of both data and procedures, does not supply the teachings missing from Nakajima. The Examiner has not shown where the MKParseDisplayName function contains methods for substituting a moniker object for the distributed object when the distributed object is streamed out from the memory. In Nakajima, MKParseDisplayName is not an stream object, but an API function of the OLE standard. As such, the MKParseDisplayName function is not a streaming object, as recited in claim 1 and as described in the subject specification. The stream object of claim 1 substitutes a moniker object for the distributed object when the distributed object is streamed out from the memory (Serial No. 09/244,291, Figs. 11-16, pp 20 *et seq*).

In addition, The Examiner has further alleged that the Nakajima teaching of extension 72 is the same as the distributed object recited in claim 1. Again, Applicant respectfully disagrees with the Examiner's analogy. In Nakajima, extension 72 is not a distributed object, but an Dynamic Link Library (DLL) according to the Microsoft Corporation's Internet Server Application Programming Interface (ISAPI) (Nakajima, column 1, line 65 to column 2, line 6). Even if the ISAPI extension is an object that consists of both data and procedures, as alleged by the Examiner and with which the Applicants respectfully disagree, that fact alone does not change the non-distributed nature of the ISAPI extension and supply the teachings missing from Nakajima. The ISAPI extension disclosed by Nakajima is distinctly different than the distributed object recited in claim 1, even if the ISAPI extension may be implemented in object oriented technology.

Accordingly, Applicant respectfully traverses the rejection of claims under 35 U.S.C. 103(a) on the grounds that the Examiner has failed to create a *prima facie* case of obviousness for : i) failing to show how the prior art reference teach or suggest *all* of the claim limitations, and ii) failing to show how some suggestion or motivation to combine the Atkinson and Nakajima references (MPEP §2143.03). Claims 2-10 include

3

all the limitations of claim 1 and are likewise believed allowable for at least the same reasons as claim 1.

Claims 11 and 21 includes limitation language similar to claim 1, (claim 11, lines 6-8; claim 21, lines 6-8),   Accordingly, claims 11 and 21 are   believed patentable over the teachings of Atkinson and Nakajima for at least the same reasons as claim 1. Claims 12-20 include all the limitations of claim 11 and are likewise believed allowable for at least the same reasons as claim 11.   Claim 21 and its respective dependent claims are believed patentable over the teachings of Atkinson and Nakajima whether considered singularly or in combination with any other art of record, for at least the same reasons as claim 1.

In addition, in response to the arguments set forth by the Applicant in the previous response, the Examiner has indicated that the Applicant is arguing limitations not found in the claims. Specifically, the Examiner indicated that the local memory and universal identifier are not recited in the claims, and, therefore, can not be relied upon to distinguish the pending claims over Atkinson and Nakajima.

Applicants respectfully disagree with the Examiner's position. Apparatus claim 1, as filed, recites " a moniker object which contains *an identifier that universally identifies an instance of the distributed object and a moniker name*" (claim 1, lines 4-5). Further, method claim 11, as filed, recites " instantiating a moniker object which contains *an identifier that universally identifies an instance of the distributed object and a moniker name*" (claim 11, lines 4-5). Computer program product claim 21 similarly recites computer usable medium having computer readable program code including "class code for instantiating a moniker object which contains *an identifier that universally identifies an instance of the distributed object and a moniker name* (claim 21, lines 4-7). As such Applicants respectfully assert that the arguments set forth in the prior response are adequately supported by limitations in the claims, as filed, without further modification.
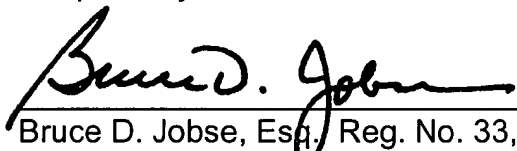
In addition, the preambles of claims 1, 11 and 21 recite an apparatus, method and computer program product, respectively, for use with a computer system having a memory, *a local storage* and an existing directory service operating in the memory (claim 1, lines 1-2; claim 11, lines 1-2; claim 21, lines1-2).   Again, Applicants

4

respectfully assert that the arguments set forth in the prior response are adequately supported by limitations in the claims, as filed, without further modification.

Applicant respectfully reasserts all of the arguments and traversals set forth in the last response to the extent still relevant to the outstanding rejections.

Applicant believes the claims are in allowable condition. A notice of allowance for this application is solicited earnestly. If the Examiner has any further questions regarding this amendment, he/she is invited to call Applicant's attorney at the number listed below. The Examiner is hereby authorized to charge any fees or credit any balances under 37 CFR §1.17, and 1.16 to Deposit Account No. 09-0460.

Respectfully submitted,

Date: 12/02/02

Bruce D. Jobse, Esq. Reg. No. 33,518
KUDIRKA & JOBSE, LLP
Customer Number 021127
Tel: (617) 367-4600 Fax: (617) 367-4656

27.   A computer program product according to claim 21 further comprising <u>program</u> [progrma] code for creating a runtime repository which includes a database of moniker name-object reference pairs.